IN THE CLAIMS

Please amend the claims as follows, substituting any amended claim(s) for the corresponding pending claim(s).

1. (Currently Amended) A data processor comprising:

an instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline;

a data cache capable of storing data values used by said pending instruction;

a plurality of architectural registers capable of receiving said data values from said data cache;

bypass circuitry capable of transferring a first data value from said data cache directly to a functional unit in one of said N processing stages without first storing said first data value in a destination one of said plurality of architectural registers; and

a cache refill controller capable of detecting that a cache miss has occurred at a first address associated with said first data value, receiving a missed cache line from a main memory coupled to said data processor, and causing ~~said first~~ a second data value to be transferred from said missed cache line to said functional unit to replace said first data value.

2. (Original) The data processor as set forth in Claim 1 wherein said cache refill controller is further capable of stalling said instruction execution pipeline after said cache miss by halting clock signals driving said instruction execution pipeline.

3. (Original) The data processor as set forth in Claim 2 further comprising a clock controller coupled to said cache refill controller and capable of generating said clock signals driving said instruction execution pipeline, wherein said clock controller stalls said instruction execution pipeline by halting said clock signals in response to a command from said cache refill controller.

4. (Currently Amended) The data processor as set forth in Claim 3 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit when said instruction execution pipeline is stalled.

5. (Original) The data processor as set forth in Claim 4 wherein said cache refill controller is further capable of storing said missed cache line into said data cache.

6. (Currently Amended) The data processor as set forth in Claim 5 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit by retrieving said ~~first~~ second data value from said missed cache line stored in said data cache.

7. (Currently Amended) The data processor as set forth in Claim 6 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit after said cache miss via said bypass circuitry.

8. (Currently Amended) The data processor as set forth in Claim 7 wherein said clock controller generates an early clock signal when said execution pipeline is stalled, wherein said early clock signal causes said ~~first~~ second data value to be transferred to said functional unit from said data cache.

9. (Original) The data processor as set forth in Claim 8 wherein said cache refill controller restarts said instruction execution pipeline after said clock controller generates said early clock signal.

10. (Currently Amended) A processing system comprising:

a data processor;

a memory coupled to said data processor;

a plurality of memory-mapped peripheral circuits coupled to said data processor for performing selected functions in association with said data processor, wherein said data processor comprises:

an instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline;

a data cache capable of storing data values used by said pending instruction;

a plurality of architectural registers capable of receiving said data values from said data cache;

bypass circuitry capable of transferring a first data value from said data cache directly to a functional unit in one of said N processing stages without first storing said first data value in a destination one of said plurality of architectural registers; and

a cache refill controller capable of detecting that a cache miss has occurred at a first address associated with said first data value, receiving a missed cache line from

a main memory coupled to said data processor, and causing ~~said first~~ a second data value to be transferred from said missed cache line to said functional unit to replace said first data value.

11. (Original) The processing system as set forth in Claim 10 wherein said cache refill controller is further capable of stalling said instruction execution pipeline after said cache miss by halting clock signals driving said instruction execution pipeline.

12. (Original) The processing system as set forth in Claim 11 further comprising a clock controller coupled to said cache refill controller and capable of generating said clock signals driving said instruction execution pipeline, wherein said clock controller stalls said instruction execution pipeline by halting said clock signals in response to a command from said cache refill controller.

13. (Currently Amended) The processing system as set forth in Claim 12 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit when said instruction execution pipeline is stalled.

14. (Original) The processing system as set forth in Claim 13 wherein said cache refill controller is further capable of storing said missed cache line into said data cache.

15. (Currently Amended) The processing system as set forth in Claim 14 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit by retrieving said ~~first~~ second data value from said missed cache line stored in said data cache.

16. (Currently Amended) The processing system as set forth in Claim 15 wherein said cache refill controller causes said ~~first~~ second data value to be transferred to said functional unit after said cache miss via said bypass circuitry.

17. (Currently Amended) The processing system as set forth in Claim 16 wherein said clock controller generates an early clock signal when said execution pipeline is stalled, wherein said early clock signal causes said ~~first~~ second data value to be transferred to said functional unit from said data cache.

18. (Original) The processing system as set forth in Claim 17 wherein said cache refill controller restarts said instruction execution pipeline after said clock controller generates said early clock signal.

19. (Currently Amended) For use in a data processor comprising 1) an instruction execution pipeline comprising N processing stages, each of the N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by the instruction execution pipeline, and 2) bypass circuitry capable of transferring a first data value from a data cache directly to a functional unit in one of the N processing stages without first storing the first data value in a destination architectural register, a method of handling a cache miss comprising the steps of:

detecting that a cache miss has occurred at a first address associated with the first data value;

stalling the operation of the instruction execution pipeline;

receiving a missed cache line from a main memory coupled to the data processor;

transferring ~~the first~~ a second data value from the missed cache line to the functional unit to replace the first data value.

20. (Original) The method as set forth in Claim 19 wherein the step of stalling comprises the sub-step of halting clock signals driving the instruction execution pipeline.

21. (Original) The method as set forth in Claim 20 further comprising the step of storing the missed cache line into the data cache.

22. (Currently Amended) The method as set forth in Claim 21 wherein the step of transferring comprises the sub-step of retrieving the ~~first~~ second data value from the missed cache line stored in the data cache.

23. (Currently Amended) The method as set forth in Claim 23 wherein the step of transferring further comprises the sub-step of transferring the ~~first~~ second data value to the functional unit after the cache miss via the bypass circuitry.

24. (Currently Amended) The method as set forth in Claim 23 further comprising the step of restarting the instruction execution pipeline after completion of the sub-step of transferring the ~~first~~ second data value to the functional unit after the cache miss via the bypass circuitry.